

Notice of Allowability

Application No.

10/723,750

Examiner

Thanh-Ha Dang

Applicant(s)

BECKER-SZENDY ET AL.

Art Unit

2163

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 02/09/07.
2. ☒ The allowed claim(s) is/are 1-4, 6-12, 14, 16-22, 24-34 and 36-40 are allowed and are renumbered as 1-35.
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 - a) ☐ All b) ☐ Some* c) ☐ None of the:
 1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

* Certified copies not received: _____.


Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
 - (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
 - 1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
 - (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08),
Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit
of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☒ Interview Summary (PTO-413),
Paper No./Mail Date 022607.
7. ☒ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.


DON WONG
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

DETAILED ACTION.

Response to Amendment

1. Receipt of Applicant's Amendment filed 02/09/07 is acknowledged.

EXAMINER'S AMENDMENT

2. Authorization for this examiner's amendment was given in a telephone interview with Mr. Samuel A. Kassatly (RN 32,247) on 02/23/07.

An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

The application has been amended as follows:

1. (Currently amended) A processor-implemented method of federating a local file system into a distributed file system while preserving local access to an existing data in the local file system, comprising:

adding a federation layer that allows both a local client and a plurality of distributed clients to access the existing data;

allowing local applications to access both the existing data in the local file system and data in other parts of the distributed file system;

wherein the federation layer establishes a detour between the local applications and the local file system, to provide access to the distributed file

system ;

~~assigning a unique object ID number to each object found in the local file system;~~

generating ~~the~~ a unique object ID number for each object found in the local file system, as a tuple $\langle I, G \rangle$ of an inode number of the object found in the local file system $[I,]$ and a generation number;

wherein the generation number is increased anytime a same inode number is found for a different object;

wherein the federation layer contains a database that stores the unique object ID number and that maps a file name and a corresponding unique object ID number; and

maintaining a deleted flag in the database for objects that have been deleted, so the occurrence of the same inode number for different objects ~~can be~~ are recognized.

2. (Original) The method of claim 1, wherein the federation layer comprises a virtual server that serves file data and metadata from the local file system to the distributed file system.

3. (Currently amended) The method of claim 2, wherein the virtual server comprises:

a virtual metadata server that serves file metadata;

a virtual storage server that serves file content; and

wherein the virtual metadata server and the virtual storage server serve data from the local file system to the distributed file system.

4. (Original) The method of claim 3, wherein the virtual storage server is a virtual object storage server.

5. (Canceled)

6. (Original) The method of claim 2, wherein the federation layer contains a distributed file system client for transferring and translating communications from a local application on the local computer system to the virtual server.

7. (Currently amended) The method of claim 2, further comprises installing the ~~file-virtual~~ server in the local ~~computer-file~~ system.

8. (Currently amended) The method of claim 3, further comprises installing the virtual metadata server in the local ~~computer-file~~ system.

9. (Currently amended) The method of claim 4, further comprises installing the virtual object storage server in the local ~~computer-file~~ system.

10. (Currently amended) The method of claim 6, further comprises installing the

distributed file system client in the local ~~computer~~file system.

11. (Currently amended) The method of claim 2, further comprises configuring the ~~file~~virtual server to communicate with the distributed file system.

12. (Currently amended) The method of claim 11, further comprises:

temporarily disconnecting the local applications on the local ~~computer~~file system; and

reconfiguring the local applications to communicate with the distributed file system ~~client~~.

13. (Canceled)

14. (Previously presented) The method of claim 1, further comprises generating the unique object ID number by counting the objects that were found in the local file system.

15. (Canceled)

16. (Currently amended) A processor-implemented method of federating a local file system into a distributed file system while preserving local access to an existing data in the local file system, comprising:

adding a federation layer that allows both a local client and a plurality of distributed clients to access the existing data;

allowing local applications to access both the existing data ~~exposed~~ in the local file system and data in other parts of the distributed file system;

wherein the federation layer establishes a detour between the local applications and the local file system, to provide access to the distributed ~~mechanism~~ file system;

wherein the federation layer contains an object ID database for mapping between a file name and a unique object ID number;

assigning the unique object ID number to each object found in the local file system;

detecting multiple hard links to a same file by comparing an inode number of files;

if upon determination that one file has more than one hard link, using the same unique object ID number for the multiple names for that object; and

if upon determination that the file has one hard link that has been removed, then deleting the object when the one hard link has been unlinked.

17. (Original) The method of claim 6, further comprises using a shared memory between the distributed file system client and the virtual server to enhance communication between the distributed file system client and the virtual server.

18. (Previously presented) The method of claim 6, further comprises moving the virtual server into a kernel space for efficiency purpose.

19. (Currently amended) A computer program product having instruction codes stored on a computer-usable storage medium for federating a local file system into a distributed file system while preserving local access to an existing data in the local file system, comprising:

a set of instruction codes for adding a federation layer that allows both a local client and a plurality of distributed clients to access the existing data;

a set of instruction codes for allowing local applications to access both the existing data in the local file system and data in other parts of the distributed file system; and

wherein the federation layer establishes a detour between the local applications and the local file system, to provide access to the distributed file system ;

~~a set of instruction codes for assigning a unique object ID number to each object found in the local file system;~~

a set of instruction codes for generating the a unique object ID number for each object found in the local file system, as a tuple $\langle I, G \rangle$ of an inode number of the object found in the local file system $[I,]$ and a generation number;

a set of instruction codes for increasing the generation number anytime a same inode number is found for a different object;

wherein the federation layer contains a database that stores the unique object ID number and that maps a file name and a corresponding unique object ID number;

the database storing a deleted flag for objects that have been deleted, so the occurrence of the same inode number for different objects ~~can be~~ are recognized; and

a user interface for providing the local access to the existing data in the local file system, using the deleted flag in the database.

20. (Original) The computer program product of claim 19, wherein the federation layer comprises a virtual server that serves file data and metadata from the local file system to the distributed file system.

21. (Currently amended) The computer program product of claim 20, wherein the virtual server comprises:

a virtual metadata server that serves file metadata;

a virtual storage server that serves file content; and

wherein the virtual metadata server and the virtual storage server serve data from the local file system to the distributed file system.

22. (Original) The computer program product of claim 21, wherein the virtual storage server is a virtual object storage server.

23. (Canceled)

24. (Currently amended) The computer program product of claim 20, wherein the federation layer contains a distributed file system client for transferring and translating communications from a local application on the local ~~computer~~file system to the virtual server.

25. (Currently amended) The computer program product of claim 20, further comprises a ~~third~~ set of instruction codes for installing the a file server in the local ~~computer~~file system.

26. (Currently amended) The computer program product of claim 21, further comprises a set of instruction codes for installing the virtual metadata server and the a virtual file in the local ~~computer~~file system.

27. (Currently amended) The computer program product of claim 22, further comprises a set of instruction codes for installing the virtual object storage server in the local ~~computer~~file system.

28. (Currently amended) The computer program product of claim 22, further comprises a set of instruction codes for installing the distributed file system ~~client~~

in the local ~~computer~~file system.

29. (Currently amended) The computer program product of claim 20, further comprises a set of instruction codes for configuring ~~the~~a file server to communicate with the distributed file system.

30. (Currently amended) The computer program product of claim 29, further comprises:

- a set of instruction codes for temporarily disconnecting the local applications on the local ~~computer~~file system; and

- a set of instruction codes for reconfiguring the local applications to communicate with the distributed file system ~~client~~.

31. (Currently amended) A ~~processor-implemented service~~ computer system for federating a local file system into a distributed file system while preserving local access to an existing data in the local file system, comprising:

- an addition of a federation layer that allows both a local client and a plurality of distributed clients to access the existing data;

- an allowance of local applications to access both the existing data in the local file system and data in other parts of the distributed file system;

- wherein the federation layer establishes a detour between the local applications and the local file system, to provide access to the distributed file

system ;

~~an assignment of a unique object ID number to each object found in the local file system;~~

a generation of the a unique object ID number for each object found in the local file system, as a tuple $\langle I, G \rangle$ of an inode number of the object found in the local file system $[I,]$ and a generation number;

wherein the generation number is increased anytime a same inode number is found for a different object;

wherein the federation layer contains a database that stores the unique object ID number and that maps a file name and a corresponding unique object ID number;

wherein the database stores a deleted flag in the database for objects that have been deleted, so the occurrence of the same inode number for different objects ~~can be~~ are recognized; and

a user interface for providing the local access to the existing data in the local file system, using the deleted flag in the database.

32. (Currently amended) The ~~service~~ computer system of claim 31, wherein the federation layer comprises a virtual server that serves file data and metadata from the local file system to the distributed file system.

33. (Currently amended) The ~~service~~ computer system of claim 32, wherein the

virtual server comprises:

a virtual metadata server that serves file metadata;

a virtual storage server that serves file content; and

wherein the virtual metadata server and the virtual storage server serve data from the local file system to the distributed file system.

34. (Currently amended) The ~~service~~ computer system of claim 33, wherein the virtual storage server is a virtual object storage server.

35. (Canceled)

36. (Currently amended) The ~~service~~ computer system of claim 32, wherein the federation layer contains a distributed file system client for transferring and translating communications from a local application on the local ~~computer-file~~ system to the virtual server.

37. (Currently amended) The ~~service~~ computer system of claim 32, further comprises an installation of the ~~a~~ file server in the local ~~computer-file~~ system.

38. (Currently amended) The ~~service~~ computer system of claim 33, further comprises an installation of the virtual metadata server and the ~~a~~ file server in the local ~~computer-file~~ system.

39. (Currently amended) The ~~service~~ computer system of claim 34, further comprises an installation of the virtual object storage server in the local ~~computer~~ file system.

40. (Currently amended) The ~~service~~ computer system of claim 36, further comprises an installation of the distributed file system ~~client~~ in the local ~~computer~~ file system.

41 - 42. (Canceled)

Allowable Subject Matter

3. Claims 1-4, 6-12, 14, 16-22, 24-34 and 36-40 are allowed, and are renumbered as 1-35.

The following is an examiner's statement of reasons for allowance: Claims 1-4, 6-12, 14, 16-22, 24-34 and 36-40 are allowable because the prior art made of record does not teach or fairly suggest the combination of elements as recited in independent Claims 1, 16, 19 and 31.

Specifically, the prior art of record does not teach:

- generating a unique object ID number for each object found in the local file system, as a tuple <I,G> of an inode number of the object found in the local file system and a generation number taken with the other limitations as recited in Claim 1.

- detecting multiple hard links to a same file by comparing an inode number of files taken with the other limitations as recited in Claim 16.
- a set of instruction codes for generating a unique object ID number for each object found in the local file system, as a tuple $\langle I, G \rangle$ of an inode number of the object found in the local file system and a generation number taken with the other limitations as recited in Claim 19.
- a generation of a unique object ID number for each object found in the local file system, as a tuple $\langle I, G \rangle$ of an inode number of the object found in the local file system and a generation number taken with the other limitations as recited in Claim 31.

The dependent claims being definite, further limiting and fully enabled by the Specification are also allowed.

These features, together with the other limitations of the independent claims are novel and non-obvious over the prior art of record.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Contact Information

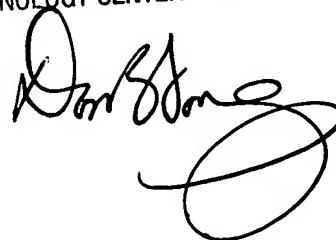
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Thanh-Ha Dang whose telephone number is 571-272-4033. The examiner can normally be reached on Monday-Friday from 9:00 AM to 5:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Don Wong can be reached on 571-272-1834. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Thanh-Ha Dang
Examiner
Art Unit 2163

DON WONG
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

A handwritten signature in black ink, appearing to read "Don Wong", with a large, stylized loop at the end.